

Certificate of Facsimile  
I hereby certify that this correspondence is being  
transmitted electronically to the U.S. Patent and  
Trademark Office June 8, 2007  
(Date of Transmission)

FRANK C. NICHOLAS (33,983)  
Name of applicant, assignee or registered representative

/FRANK C. NICHOLAS/  
Signature

June 8, 2007  
Date of Signature

PATENT  
**Case No.: AUS920000801US1**  
(9000/11)

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re patent application of:	)	
	)	
STEVEN M. FRENCH, ET AL.	)	Examiner: NGUYEN, THANH
	)	
Serial No.: 09/731,624	)	
	)	
Filed: DECEMBER 7, 2000	)	Group Art Unit: 2144
	)	
Title: METHOD AND SYSTEM FOR	)	
GENERATING A LIST OF OPERATING	)	
SYSTEMS FOR A TARGET DEVICE	)	

**SUPPLEMENTAL APPEAL BRIEF**

Mail Stop Appeal Briefs - Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Dear Sir:

Appellants herewith respectfully elect to maintain their appeal and present their  
supplemental appeal brief as follows:

**TABLE OF CONTENTS**

1.	Real Party In Interest . . . . .	3
2.	Related Appeals And Interferences . . . . .	4
3.	Status Of Claims . . . . .	5
4.	Status Of Amendments . . . . .	6
5.	Summary Of Claimed Subject Matter. . . . .	7
6.	Grounds Of Rejection To Be Reviewed On Appeal . . . . .	9
7.	Arguments . . . . .	10
8.	Summary . . . . .	15
	Claims Appendix . . . . .	16
	Evidence Appendix . . . . .	22
	Related Proceedings Appendix . . . . .	22

1. REAL PARTY IN INTEREST

The real party in interest remains Assignee INTERNATIONAL BUSINESS MACHINES CORPORATION, by virtue of an assignment executed by the inventors on December 4 and 5, 2000 and filed with the United States Patent and Trademark Office on December 7, 2000, recorded at reel number 011372 frame number 0108.

2. RELATED APPEALS AND INTERFERENCES

Appellants and the undersigned attorneys are not aware of any appeals or any interferences which will directly affect or be directly affected by or having a bearing on the Board's decision in the pending appeal.

3. STATUS OF CLAIMS

Claims 12-20 stand rejected as anticipated by Beelitz, et. al., United States Patent No. 6,182,275B1.

Claims 1-11 and 21-27 stand rejected as unpatentable over Beelitz in view of DeSimone et. al, United States Patent 5,905,872.

Claims 1-27 are the claims on appeal. *See*, Appendix.

4. STATUS OF AMENDMENTS

Claims 24-27 were added and entered into the application in the response to the May 14, 2004 office action. Claims 1, 12, and 21 were each amended in the same response.

## 5. SUMMARY OF CLAIMED SUBJECT MATTER

In this summary of claimed subject matter, all citations are to the specification of United States Patent Application 09/731,624 filed on December 7, 2000. Further, all citations are illustrative only and support for the cited element may be found elsewhere in the specification.

The invention relates to a method of dynamically creating a list of operating systems for a target device in communication with a server, the target device to be remotely booted by the server, prior to executing an operating system on the target device (p. 3, lines 14-16). The method receives from the server, at the target device, an available operating systems list of at least one operating system available to the target device (p.3, lines 16-17). The method further determines a hardware configuration of the target device and determines if the hardware configuration is compatible with each operating system from the available operating systems list (p. 3, lines 17-19). A compatible operating systems list is generated (p. 3, lines 19-20).

Specifically, Claim 1 recites a method of dynamically creating a list of operating systems for a target device in communication with a server, the target device to be remotely booted by the server, prior to executing an operating system on the target device. The method includes receiving from the server, at the target device, an available operating systems list of at least one operating system available to the target device (p. 3, lines 14-17; FIG. 4, 414) and determines a hardware configuration of the target device and whether the hardware configuration is compatible with each operating system from the available operating systems list (p. 3, lines 17-19, FIG. 4, 424). Additionally, the method generates a compatible operating systems list (p. 3, lines 19-20, FIG. 4, 448).

Claim 12 recites a computer program product in a computer usable medium for dynamically creating a list of operating systems for a target device in communication with a server, the target device to be remotely booted by the server. The product includes means for includes receiving from the server, at the target device, an available operating systems list of at least one operating system available to the target device (p. 3, lines 14-17, FIG. 4, 414) and determines a hardware configuration of the target device and whether the hardware configuration is compatible with each operating system from the

available operating systems list (p. 3, lines 17-19, FIG. 4, 424). Additionally, the method generates a compatible operating systems list (p. 3, lines 19-20, FIG. 4, 448).

Claim 21 recites a network data processing system including means for sending an available operating systems list from a server to a target device (FIG. 4, 446), the target device to be remotely booted by the server, prior to executing an operating system on the target device. The system further includes means for determining a hardware configuration of the target device (FIG. 4, 424), means for determining if the hardware configuration is compatible with each operating system from the available operating systems list (446), and means for generating a compatible operating systems list (FIG. 4, 448).

Claim 24 recites a method of dynamically creating a list of operating systems for a target device in communication with a server, the target device to be remotely booted by the server, prior to executing an operating system on the target device. The method includes receiving from the server, at the target device, an available operating systems list of at least one operating system available to the target device (p. 3, lines 14-17, FIG. 4, 414). Additionally, the method includes determining a hardware configuration of the target device and determining if the hardware configuration is compatible with each operating system from the available operating systems list (p. 3, lines 17-19, FIG. 4, 424). Furthermore, the method includes generating a compatible operating systems list (p. 3, lines 19-20, FIG. 4, 448). In addition, the method includes selecting a preferred operating system for the target device from the compatible operating systems list and executing the preferred operating system on the target device (FIG. 4, 452-460).



6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 12-20 were rejected under 35 U.S.C. §102(e) under Beelitz.

Claims 1-11 and 21-27 were rejected as unpatentable over Beelitz in view of DeSimone et. al, United States Patent 6,182,275.

7. ARGUMENTS

**A. The Examiner rejected claims 12-20 as anticipated under 35 U.S.C. §102(e) by Beelitz**

The §102(e) rejection of claims 12-20 was traversed. In order to maintain this §102(e) rejection, each and every element of the claimed invention must be disclosed in as great detail by the reference. Because the reference does not disclose each and every element, this rejection must fall. Appellant notes that this argument is quite similar to the argument presented on the appeal brief.

Claim 12 requires, *inter alia*, “means for receiving at the target device an available operating systems list of at least one operating system available to the target device.” Contrary to the Examiner’s assertions, Beelitz does not disclose such an element. Specifically, Beelitz, at most, discloses “[I]n step 207, control 103 *provides to the user interface 105* a list of the operating system types available.” (emphasis added) at column 7 55-56. Furthermore, “[I]n step 204, *control 103 accesses the master data base 125* to create or generate a list of operating system types available for the targeted computer system” (column 7 lines 35-37).

Control 103 “receives an indication *from a user interface 105* indicating the desire to purchase a computer system.” Column 7, lines 30-32 (emphasis added). Thus, control 103 is distinguished from the targeted computer system 137 (FIG. 1 of Beelitz) and the terminal or user interface 105 (FIG. 1 of Beelitz).

Thus, Beelitz does not disclose “means for receiving at the target device an available operating systems list of at least one operating system available to the target device”, and instead discloses that the “operating systems list” is received at a *user interface 105*.

Indeed, such a teaching unequivocally teaches away from the instant invention. As previously noted, Beelitz is addressed to a problem facing computer manufacturers – how to install a desired operating system on a target computer that is the subject of a pending sale of the target computer. “A system for specifying, ordering, and building a build-to-order computer system.” Abstract, Beelitz. Therefore, for Beelitz, presenting the “operating systems list” to the *target device* would be pointless, as *the user does not yet have access to the target device*. In contrast, the instant invention, directed at computer networks wherein “the target device [is] to be remotely booted by the server” as noted in the preamble, “receiving from the server, at the target device, an available operating systems list of at least one operating system available to the target device” would solve a problem facing the inventors. Namely, the claimed invention illustrates how to select a preferred operating system for the target device from the compatible operating systems list and execute the preferred operating system on the target device using a remote boot.

The Examiner’s citation to column 16, lines 35-38 (extended selection below) further illustrates the failure of Beelitz to disclose each and every element of claim 12, and further demonstrates that Beelitz teaches away from the claimed invention.

In other embodiments, a user may select other hardware components to be implemented on a targeted computer system. For example, in one embodiment, the user is presented a list of hard drives compatible with the CPU and operating system selected. In other embodiments, the user is presented with a list or lists of compatible peripheral devices. For example, the present invention may also be used by a purchaser to select a compatible type of modem to be installed in the target computer system as well as the type or types of disk drives or CD ROM drives. Other options that could be presented to a user with the present invention include various types of computer chassis, keyboards, and displays. Each hardware component and software program presented to a user would be compatible with the previous selection or selections made by the user. In some embodiments, each type of hardware component would be presented in sub menu or sub-list. Each of the entries for the additional hardware items would include associated tags which indicate compatibility with previous choices.

In the instant case, the hardware components of the target device are already known (or at least pre-determined) to the user – the user is operating the target device in their presence, and is selecting an operating system to remotely boot the target device.

Because Beelitz fails to disclose “means for receiving at the target device an available operating systems list of at least one operating system available to the target device” as claimed in claim 12, and claims 13-20 depending therefrom, this §102(e) rejection must fall.

**B. The Examiner rejected claims 1-11 and 21-27 as unpatentable over Beelitz in view of DeSimone**

In order to maintain this §103(a) rejection, each and every claim element must be taught or suggested by the references, alone or in combination, in at least as great detail as claimed. Since Beelitz in view of DeSimone fails to teach or suggest “receiving from the server, at the target device, an available operating systems list of at least one operating system available to the target device,” as claimed in claims 1 and 24 or “means for sending an available operating systems list from a server to a target device, the target device to be remotely booted by the server, prior to executing an operating system on the target device” as claimed in claim 21, this rejection must fall.

First, even if the combination of references were proper, which it is not, the combination fails to teach the claimed invention. The claims require receiving from the server, at the target device, an available operating systems list of at least one operating system available to the target device. At most, the references teach that a list is sent to a user – not a target device.

Combining Beelitz’s teachings with sending a “list of the subnetwork addresses of each client” (DeSimone, column 5, lines 47-51) fails to teach or suggest each and every element of the claims, and therefore the §103(a) rejection is flawed for at least this reason.

The Examiner's reliance on DeSimone to teach "server sends a list" is misplaced for at least two reasons. One, whether or not DeSimone teaches "server sends a list" any such teaching does not teach receiving from the server, at the target device, an available operating systems list of at least one operating system available to the target device, and the Examiner properly does not rely on Beelitz to cure such a defect. The Examiner specifically, and correctly, notes that Beelitz does not teach such an element.

Furthermore, DeSimone does not teach that the "list" taught by "server sends a list" is an "available operating systems list" as claimed. Rather, DeSimone teaches sending a list of the "subnetwork addresses of each client I to all sending clients j(j≠I) via optional headers." See, DeSimone, column 5, lines 47-51 as cited by the Examiner. Such a teaching is a far cry from the claimed elements and therefore, the references fail to teach or suggest each and every claim limitation for at least this additional reason.

Additionally, and as noted above, Beelitz unequivocally teaches away from the claims – there is no rational basis for the Examiner to assert that one of ordinary skill in the art would be motivated to modify Beelitz as suggested by the Examiner. Thus, any combination of a reference with Beelitz cannot support a rejection under §103(a).

Modification of Beelitz as suggested by the Examiner would destroy the principle of operation of Beelitz, and therefore the §103(a) rejection is improper. Even if Beelitz were so modified as to send an operating systems list from the server to a user, such actions would result in the target device (i.e., the computer being purchased by the user) sitting on an assembly line – hardly a desirable result.

Furthermore, Beelitz in view of DeSimone does not disclose "a method of dynamically creating a list of operating systems for a target device in communication with a server prior to executing an operating system on the target device," as claimed in independent claim 1. Furthermore, Beelitz does not disclose "the target device [is] to be remotely booted by the server."

Per MPEP §2111.02, the preamble of a claim is to have “the import that the claim as a whole suggests for it. In claim 1, the claim as a whole suggests the limitation that the target is a remotely booted device, and Beelitz does not disclose this element. Beelitz is addressed to a problem facing computer manufacturers – how to install operating systems on computers that are being assembled. Beelitz discloses a method of installing operating systems for target computers that will be separated from the network, shipped to an end user, and then booted. See, *inter alia*, the Abstract of Beelitz, disclosing a “system for specifying, ordering, and building a build-to-order computer system.” DeSimone teaches a method of transferring connection management information in world wide web requests and responses.

In contrast to the Beelitz disclosure, the instant case addresses problems faced by network administrators with target devices that are remotely booted on a network.

At most, Beelitz discloses that a targeted computer system 137 is initially booted up to perform the operations and instructions as per associated shell script files to load the selected programs onto its hard drive and to run the tests. In one embodiment, the selected software programs and operating systems can be down loaded and installed on the targeted computer system via the Internet. See, column 15, lines 1-7. Thus, Beelitz does not disclose “the target device to be remotely booted by the server” – Beelitz teaches only loading and installing the operating system onto the targeted computer system to boot the system.

For example, see FIG. 5, and step 520 from Beelitz, column 14, lines 30-55. Beelitz uses a data file to implement the selections per the data file on the targeted computer system. “Thus, the selected operating system, selected software programs, selected patches, selected hard drive operations, the determined always run parts, and other selections are *installed, implemented and tested* by the execution of the associated shell script files. (column 14, lines 37-42)(emphasis added) “The script processing program interprets the instructions of the shell script files and acts upon those instructions *to perform the actual instructions for the installation or implementation of the part on the computer system.*” (column 14, lines 46-50)(emphasis added).

Withdrawal of the rejections to claims 1, 21, 24 and claims 2-11, 22-23, and 25-27 depending therefrom is requested.

**SUMMARY**

The Appellants respectfully request maintenance of their appeal, and submit that claims 1-27 fully satisfy the requirements of 35 U.S.C. §§102, 103 and 112. In view of the foregoing, favorable consideration and early passage to issue of the present application is respectfully requested.

Dated: **June 8, 2007**

Respectfully submitted,  
STEVEN M. FRENCH, et al.

/FRANK C. NICHOLAS/

CARDINAL LAW GROUP  
1603 Orrington Avenue, Suite 2000  
Evanston, IL 60201  
(847) 905-7111

---

Frank C. Nicholas  
Registration No. (33,983)  
Attorney for Appellants

## 10. APPENDIX

1. A method of dynamically creating a list of operating systems for a target device in communication with a server, the target device to be remotely booted by the server, prior to executing an operating system on the target device, comprising:
  - receiving from the server, at the target device, an available operating systems list of at least one operating system available to the target device;
  - determining a hardware configuration of the target device;
  - determining if the hardware configuration is compatible with each operating system from the available operating systems list; and
  - generating a compatible operating systems list.
2. The method of claim 1 wherein the hardware configuration is determined by a network discovery process.
3. The method of claim 1, further comprising:
  - mapping a specific location of the target device.
4. The method of claim 3 further comprising:
  - determining at least one location-based operating system that is based on the specific location of the target device.



5. The method of claim 4 further comprising:  
determining if the hardware configuration is compatible with the location-based operating system.
6. The method of claim 4 wherein the compatible operating systems list does not include the location-based operating system, further comprising:  
adding the location-based operating system to the compatible operating systems list.
7. The method of claim 4, wherein the available operating systems list does not include the location-based operating system, further comprising:  
adding the location-based operating system to the available operating systems list.
8. The method of claim 1, further comprising:  
selecting, at the target device, a target operating system from the compatible operating system list.
9. The method of claim 8 further comprising:  
receiving at the target device, the target operating system.
10. The method of claim 9 further comprising:  
providing an initial bootstrap to the target device; and  
executing the initial bootstrap on the target device before the target operating system is selected.

11. The method of claim 10 further comprising:  
relocating the initial bootstrap after the target operating system is received.
12. Computer program product in a computer usable medium for dynamically creating a list of operating systems for a target device in communication with a server, the target device to be remotely booted by the server, comprising:  
means for receiving at the target device an available operating systems list of at least one operating system available to the target device;  
means for determining a hardware configuration of the target device;  
means for determining if the hardware configuration is compatible with each operating system from the available operating systems list; and  
means for generating a compatible operating systems list.
13. The program of claim 12 further comprising:  
means for discovering the hardware configuration via a network discovery process.
14. The program of claim 12 further comprising:  
means for determining at least one location-based operating system based on the location of the target device.
15. The program of claim 14 further comprising:  
means for adding the location-based operating system to the available operating systems list.

16. The program of claim 14 further comprising:  
means for determining if the hardware configuration is compatible with the location-based operating system.
17. The program of claim 16 further comprising:  
means for adding the location-based operating system to the compatible operating systems list if the hardware configuration is compatible with the location-based operating system.
18. The program of claim 12, further comprising:  
means for receiving a selection of a target operating system from the compatible operating system list; and  
means for sending the target operating system to the target device.
19. The program of claim 18 further comprising:  
means for executing a network bootstrap on the target device before the target operating system is selected.
20. The method of claim 19 further comprising:  
means for relocating the network bootstrap after the target operating system is selected.

21. A network data processing system comprising:
- means for sending an available operating systems list from a server to a target device, the target device to be remotely booted by the server, prior to executing an operating system on the target device;
  - means for determining a hardware configuration of the target device;
  - means for determining if the hardware configuration is compatible with each operating system from the available operating systems list; and
  - means for generating a compatible operating systems list.
22. The system of claim 21 further comprising:
- means for determining a specific location of the target device;
  - means for determining at least one location-based operating system that is based on the specific location; and
  - means for determining if the location-based operating system is compatible with the hardware configuration.
23. The system of claim 21 further comprising:
- means for executing a network bootstrap program on the target device before a target operating system is selected;
  - means for selecting the target operating system from the compatible operating systems list;
  - means for relocating the network bootstrap program after the target operating system is selected; and
  - means for sending the target operating system to the target device.

24. A method of dynamically creating a list of operating systems for a target device in communication with a server, the target device to be remotely booted by the server, prior to executing an operating system on the target device, comprising:

receiving from the server, at the target device, an available operating systems list of at least one operating system available to the target device;

determining a hardware configuration of the target device;

determining if the hardware configuration is compatible with each operating system from the available operating systems list; and generating a compatible operating systems list;

selecting a preferred operating system for the target device from the compatible operating systems list; and

executing the preferred operating system on the target device.

25. The method of claim 24 wherein selecting a preferred operating system for the target device comprises an automatic selection.

26. The method of claim 24 wherein selecting a preferred operating system for the target device comprises a selection by a system administrator.

27. The method of claim 24 wherein selecting a preferred operating system for the target device comprises a selection by a user.

**Evidence Appendix**

None

**Related Proceedings Appendix**

None.